

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

data = np.random.randn(100)

data = np.append(data, [8, 10, -8, -10])

df = pd.DataFrame(data, columns=['Value'])

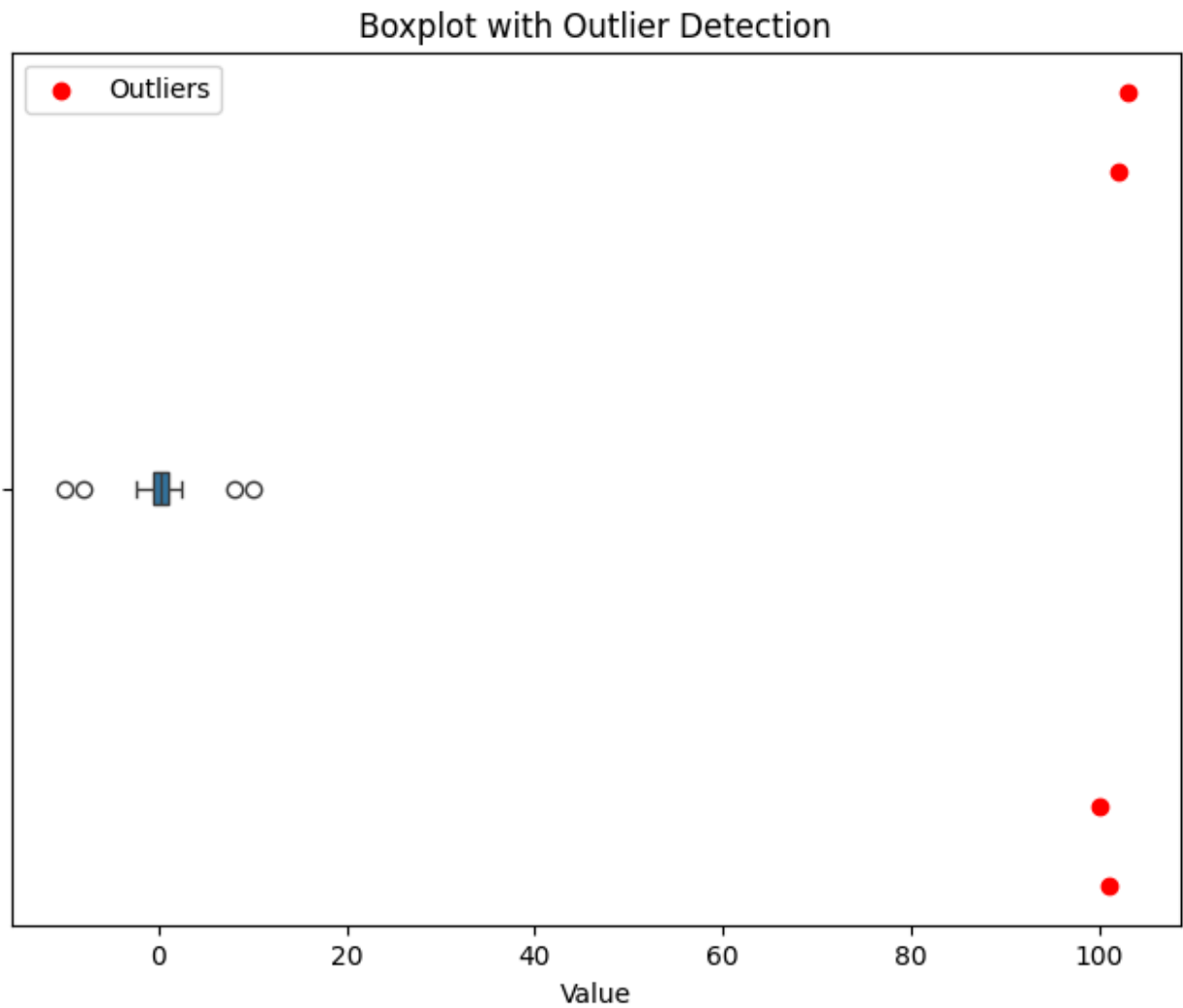
Q1 = df['Value'].quantile(0.25)
Q3 = df['Value'].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

outliers = df[(df['Value'] < lower_bound) | (df['Value'] > upper_bound)]

plt.figure(figsize=(8, 6))
sns.boxplot(x=df['Value'])
plt.scatter(outliers.index, outliers['Value'], color='red',
            label='Outliers')
plt.title('Boxplot with Outlier Detection')
plt.xlabel('Value')
plt.legend()
plt.show()

print("Outliers detected:")
print(outliers)
```



```
# prompt: z score normalization

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats

!pip install -q matplotlib-venn

# Generating normally distributed data
data = np.random.randn(500)

# Calculate IQR
Q1 = np.percentile(data, 25)
```

```

Q3 = np.percentile(data, 75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

outliers = data[(data < lower_bound) | (data > upper_bound)]

plt.figure(figsize=(10, 6))
plt.hist(data, bins=30, edgecolor='black')
plt.scatter(outliers, np.zeros_like(outliers), color='red', marker='o',
            label='Outliers')
plt.axvline(lower_bound, color='green', linestyle='--', label='Lower
Bound')
plt.axvline(upper_bound, color='green', linestyle='--', label='Upper
Bound')
plt.xlabel('Data')
plt.ylabel('Frequency')
plt.title('IQR Outlier Detection')
plt.legend()
plt.show()

data_mean = np.mean(data)
data_std = np.std(data)
normalized_data = (data - data_mean) / data_std

plt.figure(figsize=(10, 6))
plt.hist(normalized_data, bins=30, edgecolor='black')
plt.xlabel('Normalized Data')
plt.ylabel('Frequency')
plt.title('Z-Score Normalized Data')
plt.show()

```

[4]

